Requirements and Analysis Document for NNN

Table of Contents

Version: 1

Date 2014-03-27

Author: Group

This version overrides all previous versions.

1 Introduction

This section gives a brief overview of the project.

1.1 Purpose of application

We want to create a 2D Platform Fighting game in which you can be a character with attacks/skills and play against another person locally.

1.2 General characteristics of application

The application will be a desktop, standalone (non-networked), multiplayer application with a graphical user interface for the Windows/Mac/Linux platforms.

The application will be a fast paced non-turn based 2D Platform Fighting game where player can jump around on platforms and fight with other players locally.The player can damage other players and can also take damage from other players and the environment. The rounds can be time or life dependent, according to the players choice. The game ends when only one player is left alive.
The GUI will incorporate a kind of HUD that shows information about the characters during the game. There players will also be able to choose character and make other choices from a graphical menu before game.

1.3 Scope of application

Due to the limited time at our disposal we will NOT be including things like:

- -Multiplayer over INTERNET
- -Account and account progress
- -Story
- -AI
- -Shadows

1.4 Objectives and success criteria of the project

1. It should be possible to play a match of the game and win/lose against another player on the same computer. The player should be able to use one of at least two different characters.
2. The game type can be a timed game or end when only one character is left alive.

1.5 Definitions, acronyms and abbreviations

- GUI: Graphical User Interface
- HUD: Heads Up Display
- Match: A game round to decide a winner.
- AI: Artificial Intelligence

2 Requirements

2.1 Functional requirements

The player(s) should be able to:

1. Select basic options for the game
    a) Select sound sound/music to be turned on/off
    b) Select screen resolution
    c) Select standard keys for movement.
2. Start a new game
    a) Each player may choose a character and a name.
    b) Choose one game mode and a map for all players.
3. Interact in the world
    a) Move around using the chosen keys
    b) Attack either by using a weapon or a skill.
    c) Be attacked and lose HP if hit.
    d) Be affected by static and dynamic areas.
4. Win/lose a game.

2.2 Non-functional requirements

2.2.1 Usability

To start a game with basic settings should be fast and easy with a minimum amount of clicks.

Due to the game's fast-paced nature it will be geared towards getting a responsive game with fast response times and a movement system that doesn't feel "sluggish" or slow.

While in the game the user should not feel a lack of information, all relevant information should always be available in the HUD.

2.2.2 Reliability

NA

2.2.3 Performance

Any action performed by a user in game should not have a lag exceeding 50ms.
Any action performed by a user while navigating the menu should not exceed 2sec.
Loadtime of a game should be quick enough not to cause distress to user.

2.2.4 Supportability

The application must be implemented so that the GUI can easily be adapted to the webb.
No other platform than PC and PC with webb is supported.

The implementation should prepare for the dividing of the application into a client/server-architecture for net based games. It should be easy to partitioning the application into a client-server architecture.

2.2.5 Implementation

To achieve platform independence the application will use the Java environment. All hosts must have the JRE installed and configured. The application needs to be installed on all hosts where it will run (possibly downloaded).

2.2.6 Packaging and installation

The program will be delivered as executable jar file and a README with instructions.

2.2.7 Legal

There should be no copyright conflicts with the application. All elements will either be original or open source with references.

2.3 Application models

2.3.1 Use case model

See APPENDIX for UML diagram and textual descriptions.

2.3.2 Use cases priority

1. Move left/right
2. Jump/fall
3. use attack
    a. melee
    b. range
    c. range with diagonal movement
4. Move character into static area
5. Move character into dynamic area
6. Start new game
7. Start application

2.3.3 Domain model

See APPENDIX

There will be unique id's for:

- Players, the name of the player

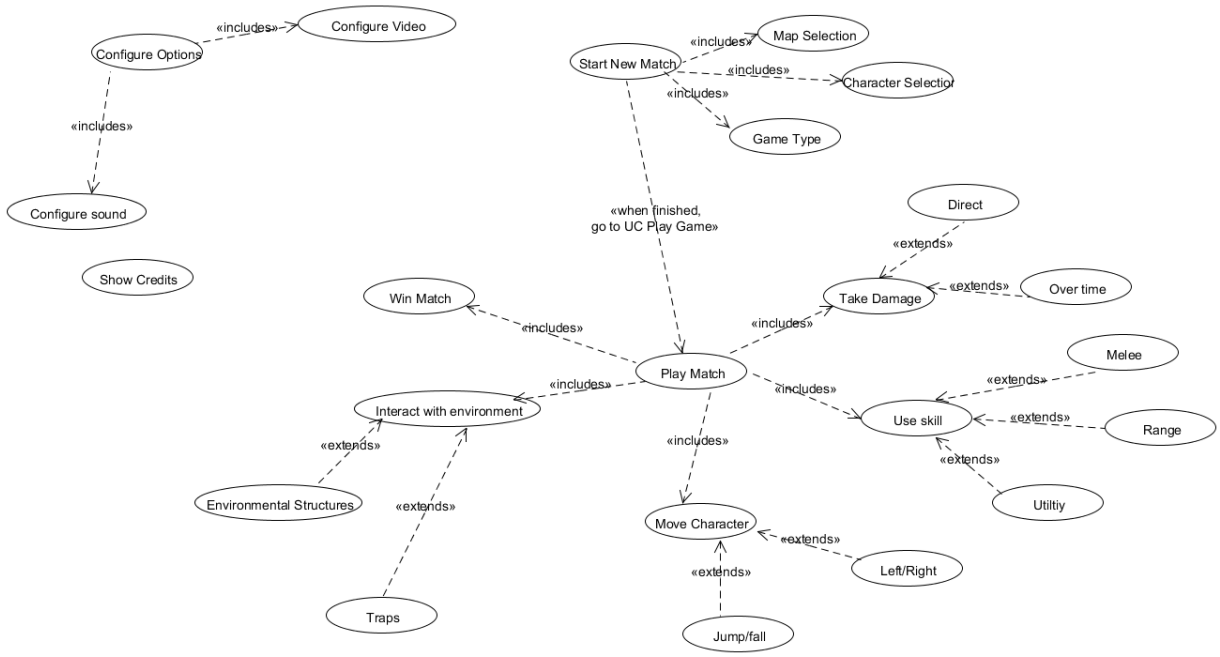
2.3.4 User interface

Text to motivate a picture.

2.4 References

APPENDIX

Use Case overview



Use case texts :

Use Case: Jump

Summary: This is how the player gets the character to jump in the game. UC Start New Match must have been executed before this UC, though not necessarily right before.
Priority: high
Extends: Move Character
Includes: Move-Fall
Participators: Actual player

Normal flow of events

A simple jump straight up with variation if user release the button early or character collides with a solid object.

	User	System
1	Player presses Jump	
2		Character image set to airborne
3		Character gets vertical speed
4		Character is affected by gravity: increasing downwards vertical speed
5	Player releases Jump	
6		If Character is on way up: stops and starts falling, otherwise no effect
7		If character hits ground with vertical speed != 0, vertical speed is set to zero.
8		Character image set to idle
5,1		Character collides impassible object
5,2		Vertical speed set to zero

Use Case: Start New Match

Summary: This is how the player moves through the New Match wizard. The application must have been launched before this UC. UC Play Match will follow this UC during normal flow of events.

Priority: medium - high

Extends: -

Includes: Select map, select character, select match type. All three are fully included here.

Participators: Actual player

Normal flow of events

The players choose game mode, characters, names and map. Then the match starts.

	User	System
1	Clicks "New Game" button	
2		Start new game wizard on step 1, showing character selection, mode selection, name entry and player count
3	Clicks "Stock"	
4		Toggles between "Time" and amount of lives. 3:00 is changed to 3
5	Clicks 1st player panel	
6		1st player selected
7	Clicks a character	
8		1st player selects the character
9	Clicks name field	
10		Name field selected
11	Writes "Kalle"	
12		1st Players name set to Kalle
13	Clicks nth player panel	repeat step 5-12 for nth player.
14	Clicks the next button	
15		Shows wizard step 2, the map selection screen
16	Clicks a map	
17		Map is selected
18	Clicks next	
19		Load characters and map, spawns players and starts match

Use Case: Use Melee Skill

Summary: This is how the player executes a melee skill in the game. UC Start New Match must have been executed before this UC, though not necessarily right before.
Priority: high
Extends: Use skill
Includes: Take damage though not explained here.
Participators: Actual player

Normal flow of events

Player presses the melee attack button and the character performs a melee attack in the direction the character is currently facing.

	User	System
1	Player presses Melee Skill key near enemy	
2		Player character image change to melee skill.
3		Player character execute melee skill in character direction.
4		Enemy character takes damage if hit.
5		HUD is updated

Use Case: Move left/right

Summary: This is how the player moves the character in the game. UC Start New Match must have been executed before this UC, though not necessarily right before.
Priority: high
Extends: Move Character
Includes: -
Participators: Actual player

Normal flow of events

A simple move left or right with no consequences.

	User	System
1	Press left/right key.	
2		Character change direction.
3		Character image align with direction.
4		Character move left/right with speed depending on the characher specification.
5		Character position change.

Use Case: Fall

Summary: This is how the player gets the character to fall in the game. UC Start New Match must have been executed before this UC, though not necessarily right before. Player must be in the air, either by jumping or by walking off a ledge.
Priority: high
Extends: Move Character
Includes: -
Participators: Actual player

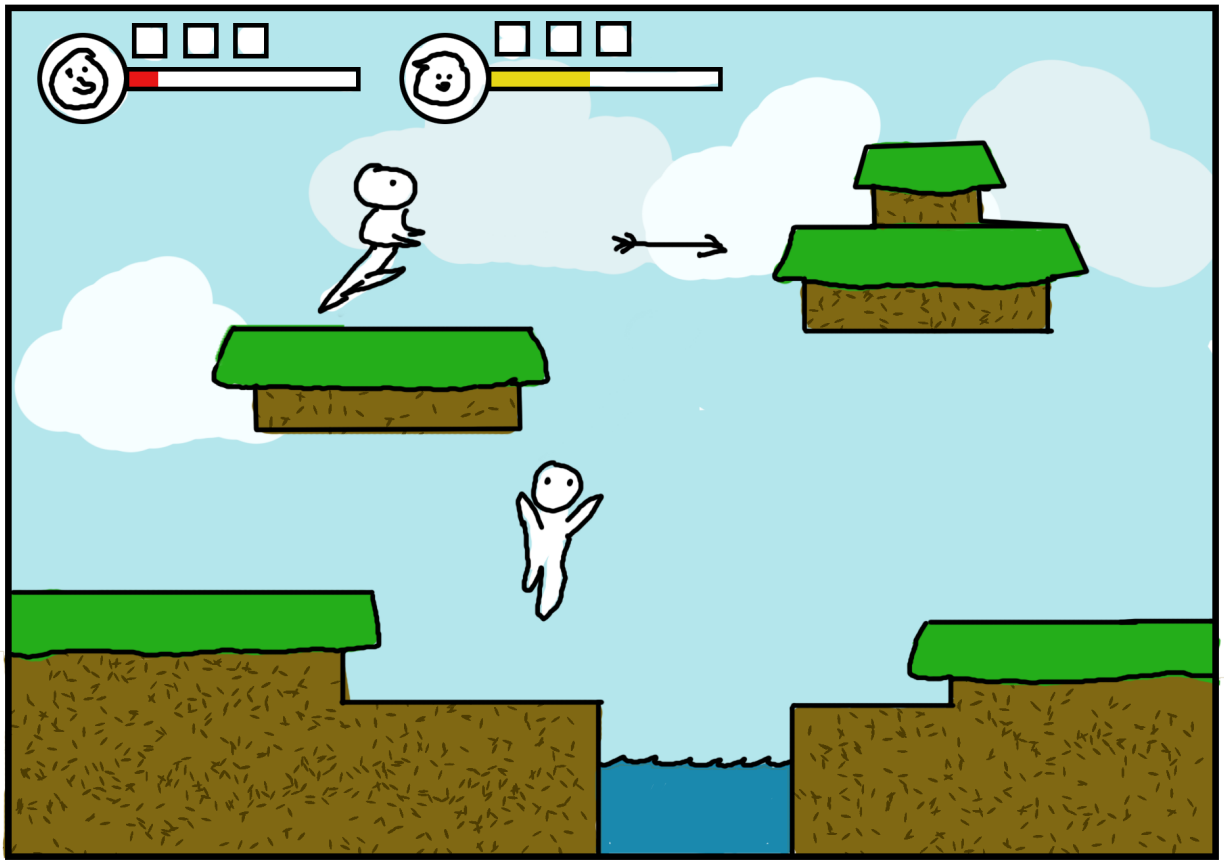
Normal flow of events

A character is in the air and accelerating towards the ground.

	User	System
1	Player falling while no side-movement.	
2		Character direction and image change to down
3		Character image change to down
4		Character move with gravitational speed
5		Character pos change

GUI

A simple GUI sketch.



Domain model